Feature selection

Feature selection

- Typical predictive analytical projects have base tables with ~ 1000 variables
- Which problems arise if we use all of these in our model?
  - Model very hard to interpret
  - Impossible to present to business
  - Unstable on long term (all variables need to be up to date)
  - Unnecessary complexity (takes longer to score model)
  - Overfitting

Feature selection

- Decision trees: variable selection is incorporated in modeling
- Other algorithms
  - Univariate selection
  - Stepwise selection

Univariate variable selection

Variable 1 | Variable 2 | Variable 3 | Variable 4 | Variable 5 | Variable 6

Build models with one variable

Model 1 | Model 2 | Model 3 | Model 4 | Model 5 | Model 6

Compute performance metric (ex: AUC)

Performance 1 | Performance 2 | Performance 3 | Performance 4 | Performance 5 | Performance 6

Select variables with performance higher than threshold

Variable 1 | **Variable 2** | Variable 3 | **Variable 4** | **Variable 5** | Variable 6

Univariate variable selection

Alternative methods

- Information gain
- Pearson correlation with target higher than threshold
- Hypothesis testing
- ...

Stepwise forward variable selection

Step 1 : Model with 1 variable

Variable 1	Variable 2	Variable 3	Variable 4	Variable 5	Variable 6

Select variable with best AUC

Variable 1	Variable 2	Variable 3	Variable 4	Variable 5	Variable 6

Stepwise forward variable selection

Step 2 : Model with 2 variables

Variable 2
Variable 1

Variable 2
Variable 3

Variable 2
Variable 4

Variable 2
Variable 5

Variable 2
Variable 6

Select variables with best AUC

Variable 2
Variable 1

Variable 2
Variable 3

Variable 2
Variable 4

Variable 2
Variable 5

Variable 2
Variable 6

Stepwise forward variable selection

Step 3 : Model with 3 variables

Variable 2
Variable 4
Variable 1

Variable 2
Variable 4
Variable 3

Variable 2
Variable 4
Variable 5

Variable 2
Variable 4
Variable 6

Select variables with best AUC

Variable 2
Variable 4
Variable 1

Variable 2
Variable 4
Variable 3

Variable 2
Variable 4
Variable 5

Variable 2
Variable 4
Variable 6

Stepwise forward variable selection

Step N : Model with all variables

Result : N models, each with 1 additional variable

Variable 2

Variable 2
Variable 4

Variable 2
Variable 4
Variable 6

Variable 2
Variable 4
Variable 6
Variable 3

Variable 2
Variable 4
Variable 6
Variable 3
Variable 1

Variable 2
Variable 4
Variable 6
Variable 3
Variable 1
Variable 5

Stepwise forward variable selection

Variable 2

AUC = 0,65

Variable 2
Variable 4

AUC = 0,72

Variable 2
Variable 4
Variable 6

AUC = 0,79

Variable 2
Variable 4
Variable 6
Variable 3

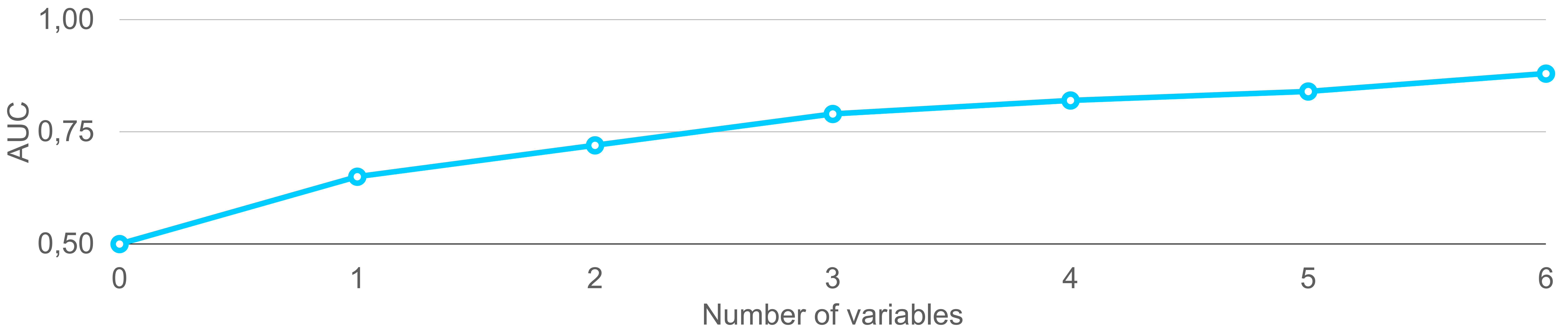
AUC = 0,82

Variable 2
Variable 4
Variable 6
Variable 3
Variable 1

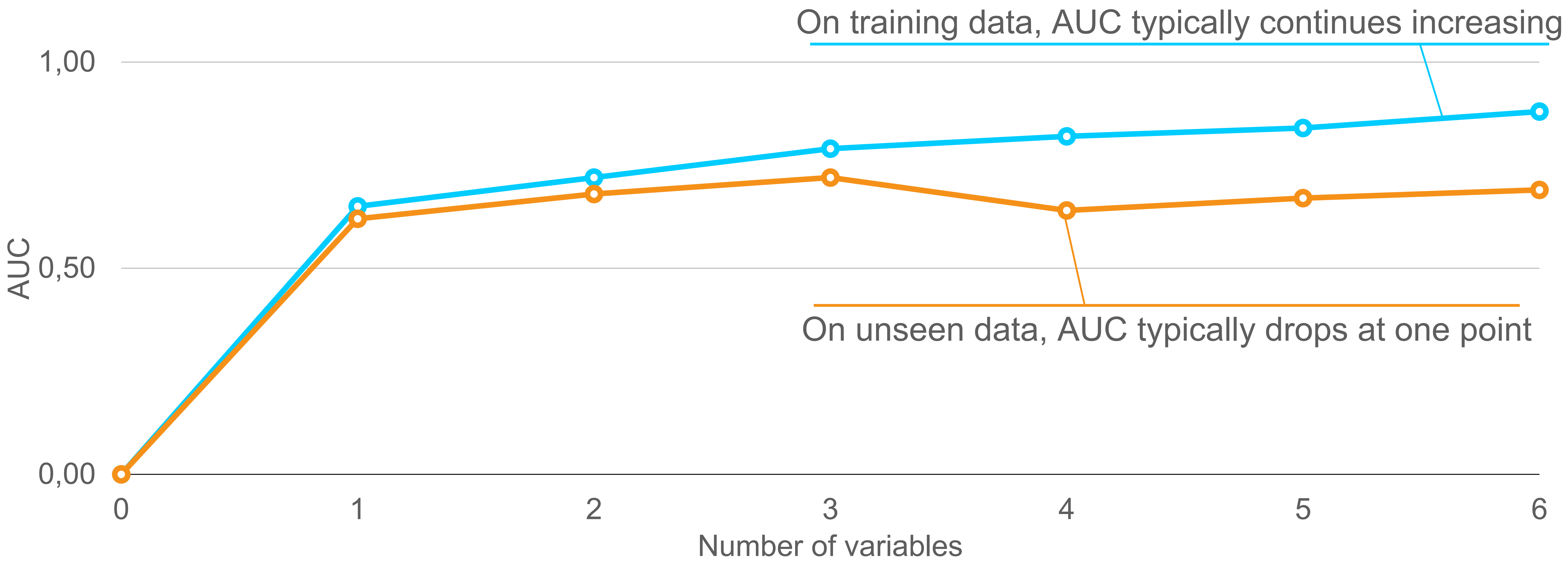
AUC = 0,84

Variable 2
Variable 4
Variable 6
Variable 3
Variable 1
Variable 5

AUC = 0,88



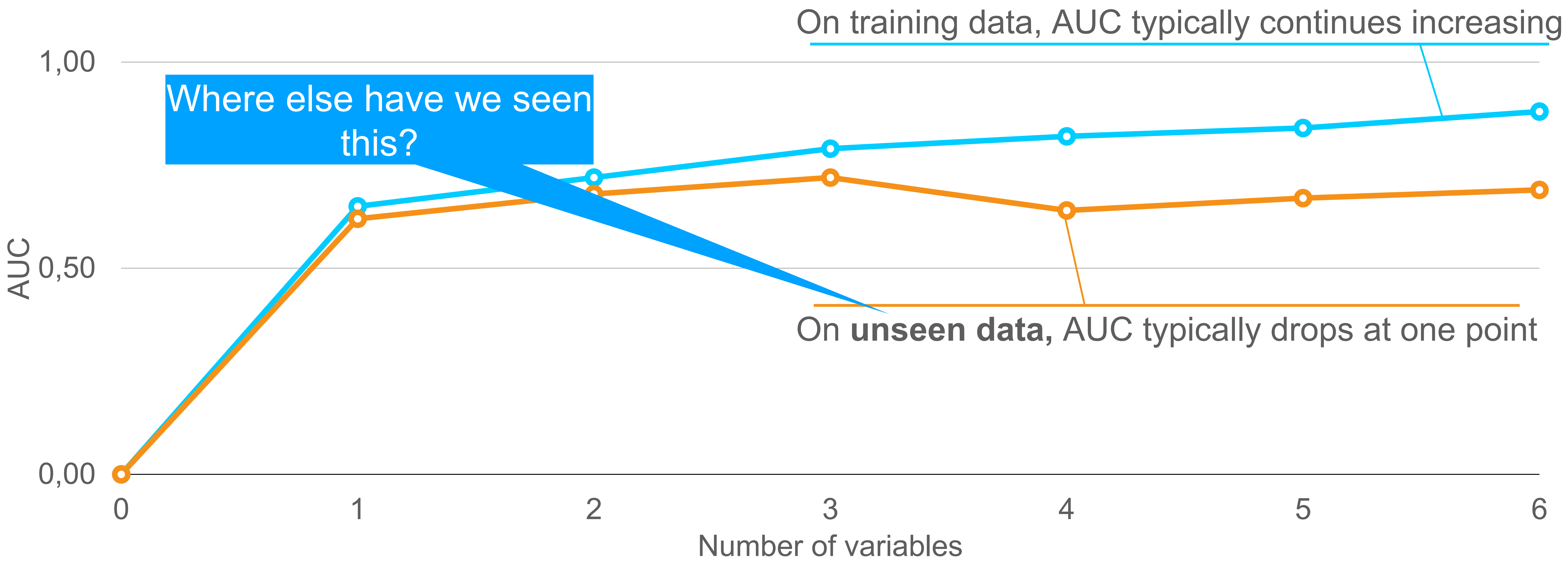
Stepwise forward variable selection



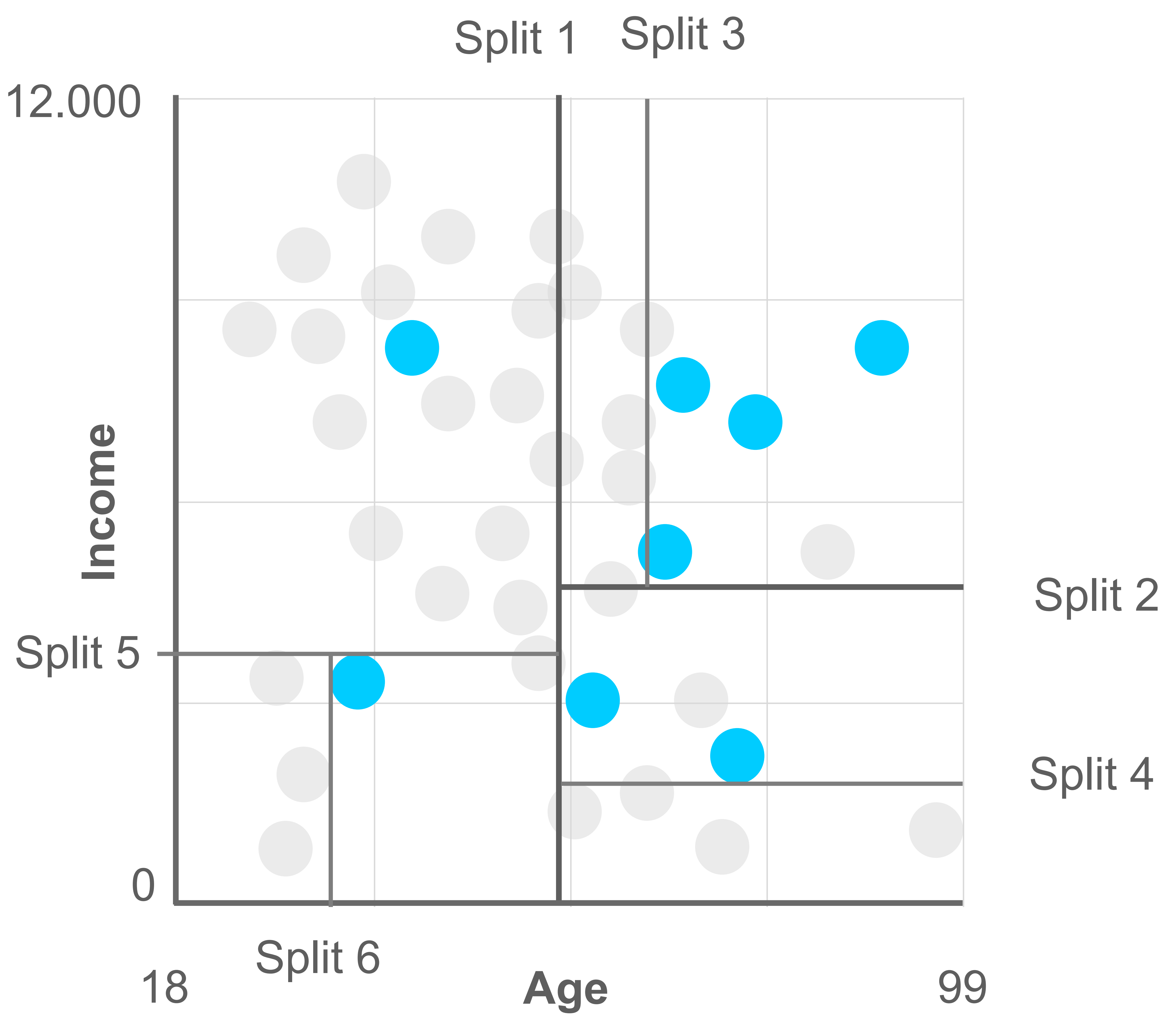
Univariate selection vs. Stepwise selection

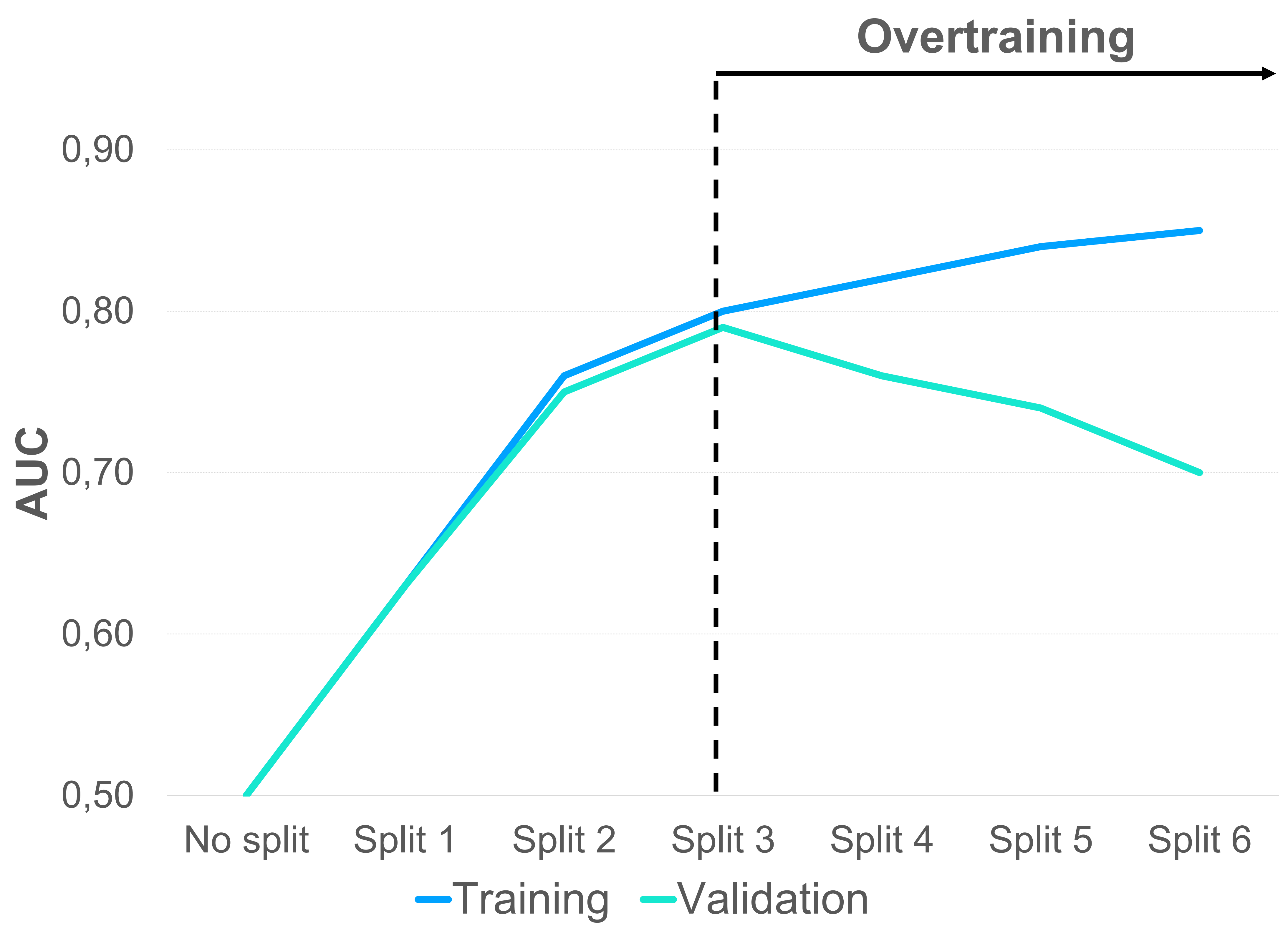
- Will the outcome of selected features be the same?

Importance of unseen data



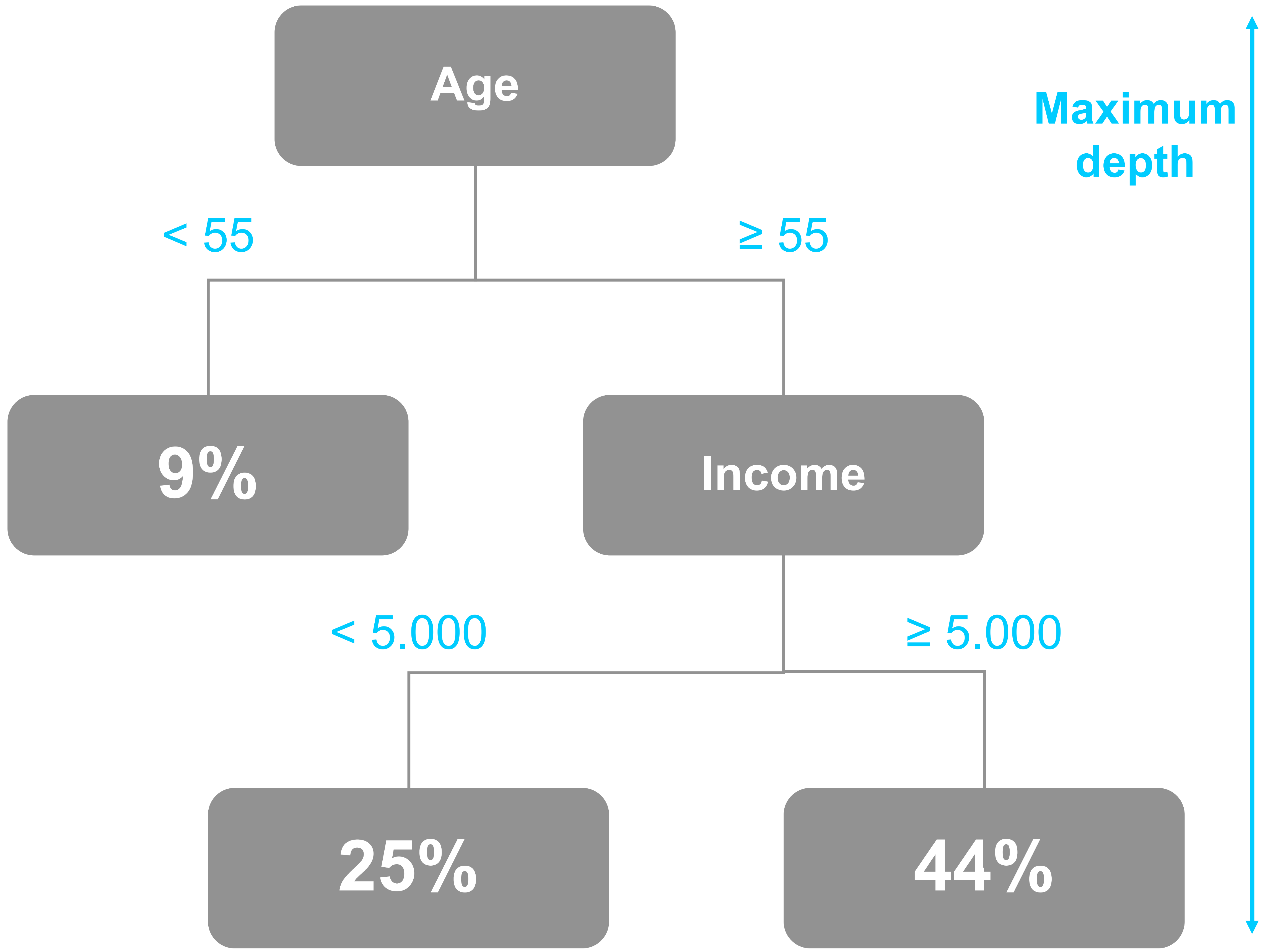
Intermezzo: let's get back to the decision tree





Pruning to mitigate this issue

Intermezzo: Decision tree



Parameters

- Optimized to minimize the impurity measurement
- Splitting decision at each branch (feature + value)

They are learned from the training data

Hyperparameters

- Configuration settings that control the behavior of the machine learning algorithm
- Maximum depth, Minimum samples per leaf,…

They are NOT learned from the training data
They are set prior to model training

Control Model Complexity
Hyperparameters can help manage the complexity of a model to **avoid overfitting or underfitting**

Intermezzo: Regulating parameters

Self study

https://scikit-learn.org/1.5/modules/generated/sklearn.svm.LinearSVC.html

LinearSVC

```
class sklearn.svm.LinearSVC(penalty='l2', loss='squared_hinge', *, dual='auto',
tol=0.0001, C=1.0, multi_class='ovr', fit_intercept=True, intercept_scaling=1,
class_weight=None, verbose=0, random_state=None, max_iter=1000)
```
[source]

penalty : *{'l1', 'l2'}, default='l2'*

Specifies the norm used in the penalization. The 'l2' penalty is the standard used in SVC. The 'l1' leads to `coef_` vectors that are sparse.

C : *float, default=1.0*

Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive. For an intuitive visualization of the effects of scaling the regularization parameter C, see Scaling the regularization parameter for SVCs.

Intermezzo: Regulating parameters

Self study

Regularization parameters play a crucial role in controlling the complexity of models and preventing overfitting across different algorithms:

Linear Regression: Regularization through Ridge, Lasso, or Elastic Net (parameters: λ\lambdaλ, λ1\lambda_1λ1, λ2\lambda_2λ2).

Decision Trees: Complexity controlled by parameters like max_depth, min_samples_split, min_samples_leaf, and max_features.

Logistic Regression: Regularization through Ridge, Lasso, or Elastic

By adjusting these regulating parameters, you can optimize your models for better performance and generalization on unseen data.